01 | 拆分与合并：如何快速地批量处理内容相似的Excel？

2021-02-03 尹会生

Python自动化办公实战课 进入课程 >

讲述：尹会生
时长 15:04 大小 13.81M

你好，我是尹会生。今天是咱们的第一节课，我先带你学习下，如何用 Python 操作 Excel。

Excel 是我们在工作中用到的最频繁的软件之一，它有着强大的计算能力和便捷的图表功能。如果我们要在同一个 Excel 文件中进行操作，手工进行也很方便，但问题是，如果我们需要同时操作多个 Excel 文件，就是一件非常耗时的事情了。

在工作场景中，需要同时操作多个 Excel 的情况主要有 2 种：批量合并和批量拆分。我来带你看 2 个场景。

批量合并。假设你需要对某些工作内容进行问卷调查，这时你用 Excel 做了调查问卷模版。我想你会这样做：先把 Excel 通过工作群分发给所有员工，再把群里收集到的反馈

附件汇总成一个文件。

批量拆分。假设你是公司的财务人员，你需要使用 Excel 对员工工资进行核算，之后再打印出来。但是公司要求员工薪水保密，所以每个员工的工资需要拆分成一个独立的文件，最后还需要打印出来。

无论是合并，还是拆分，我们都面临着一个困境：没有现成的软件可以实现多个 Excel 文件的合并和拆分操作，所以你只好对每一个 Excel 文件都进行“打开 - 复制粘贴 - 保存”的工作。

很多人在面对这样的工作需求时，都**忍不住立马去做，却很少停下来分析问题**。其实，这三步是很简单的工作，不过也是无意义的重复工作，既浪费了时间，又没有真正产生价值。

幸运的是，这些工作都可以通过 Python 来解决。今天，我就给你介绍一下用 Python 实现重复工作自动化，快速实现 Excel 的合并和拆分的方法。

如何用 Python 手工操作一个 Excel 文件?

刚刚我们说到批量处理，其实也就是逐一处理多个文件。如果我们想要提升这类工作的效率，就可以先借助 Python 把每一次处理都自动化。所以，在讲具体的合并和拆分方法前，我们要解决的第一个问题，就是用 Python 代替你的双手来操作一个 Excel 文件。

如果要用 Python 操作 Excel 文件，首先就要支持读写 Excel 的功能。**在 Python 中，要想实现对某一个功能的支持，就需要安装扩展库。**

支持 Excel 读取的扩展库叫做 xlrd 库，支持 Excel 写入的扩展库叫做 xlwt 库。我们可以使用下面的命令行进行安装：

复制代码

```
pip3 install xlrd
pip3 install xlwt
```

那么，如何使用 Python 读取 Excel 文件的内容呢？你可以使用这样的代码实现：

复制代码

```
import xlrd

file = '/Users/user1/Desktop/a.xlsx'

data = xlrd.open_workbook(file)
table = data.sheets()[0]
value = table.cell_value(rowx=4, colx=4)
```

虽然代码很简单，但是它对自动化操作 Excel 的意义非常大。

试想一下，如果能够使用 Python 替代全部的手工操作，大批量的文件就可以使用 Python 的循环功能自动化完成对每一个文件的自动处理工作了。

对于编程语言来说，文件合并的步骤可以分解为读取第一个文件，读取第二个文件，将第一个文件的内容追加到第二个文件下方。

所以在我们学会使用 Python 读取 Excel 文件之后，我们还需要掌握如何将读取的内容写入到 Excel 文件。写入文件的代码如下：

复制代码

```
import xlwt

dst_file = '/Users/edz/Desktop/文章1/result/结果.xlsx'

workbook = xlwt.Workbook(encoding='utf-8')
xlsheet = workbook.add_sheet("统计结果")

# 写入内容,假设取出的内容是value
xlsheet.write(0, 0, value)

# 保存文件
workbook.save(dst_file)
```

可以看到，写入文件的时候，我们使用了一个叫做 write 的函数。它的前两个参数代表的写入位置，分别是指定写入的行和列坐标。无需多言，这个写入位置非常重要。如果按照上面的代码方式写入，也就是前两个参数均指定为 0，就会覆盖这个 Excel 文件中的现有内容了。

所以，你如果想完成合并操作的话，就要实现对现有 Excel 内容进行追加写入。通常我们会先获取现有的内容一共有多少行、多少列，然后向后移动一个位置，再进行写入。

这种追加写入的方式，在我们的工作场景中非常常见。接下来，我们就看看怎么用追加写入的方式，实现多个 Excel 文件的合并吧。

怎样实现 Excel 的合并?

我们还是用前面提到的做调查问卷模板的场景，来具体讲一讲怎么实现 Excel 的合并。这里，我们就要用到一个重要功能了：**循环功能**。

循环功能的核心代码是：

复制代码

```
from pathlib import Path, PurePath

# 指定要合并excel的路径
src_path = '/Users/edz/Desktop/文章1/调查问卷'

# 取得该目录下所有的xlsx格式文件
p = Path(src_path)
files = [x for x in p.iterdir() if PurePath(x).match('*.xlsx')]
```

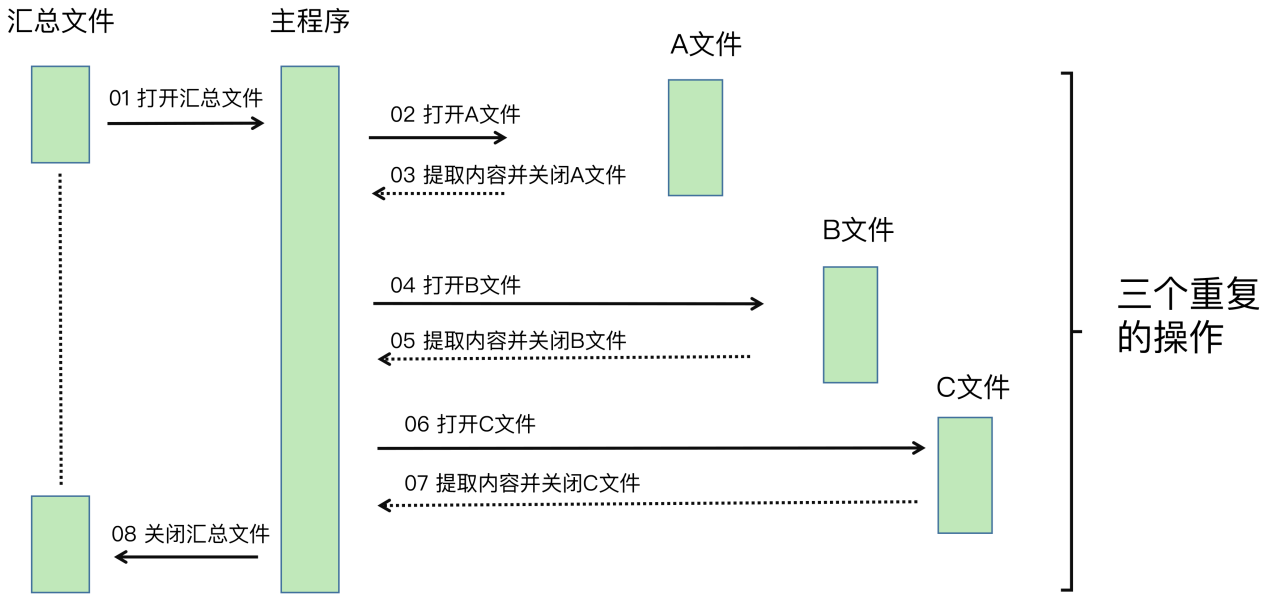
在这段代码中，我使用了 for 语句，实现了 Python 的循环功能。通过这样的功能，我可以依次获取 src_path 变量指向的路径下所有的文件。同时，为了避免这个目录里的文件类型过多，我使用一个 if 语句用于条件判断，只提取.xlsx 结尾的文件。

现在，用 Excel 实现调查问卷自动化的主要功能已经都实现了。接下来，我们看看怎样实现整个工作过程。我把它们的工作流程定义为三个步骤：

1. 找到整个工作过程当中重复操作的部分；
2. 将重复操作的部分需要哪些手工操作找出来，使用 Python 编写程序代替手工操作的部分；
3. 对重复的部分，使用循环语句进行批量处理。

如果你对把手工操作改成 Python 程序，还没有任何经验的话，我还可以为你提供更直观的办法，叫做**画时序图**。

我先带你看看时序图是什么呢？如下图表:



简单来讲，**时序图就是以时间顺序来排列程序中的事件的图表**。通过上图，你应该很容易看出重复操作的这四个事件：

1. 打开文件；
2. 提取用户填写内容；
3. 粘贴到汇总文件；
4. 关闭文件。

下面，我们就用 Python 来替代这四个事件。由于它们是重复的，所以我会使用 for 循环对它们依次进行处理。

回到我们的 Excel 做调查问卷的场景。当我们回收了调查问卷之后，每份问卷的格式是完全相同的，刚好可以利用上面提到的循环功能处理每份问卷。而问卷的选项则是我们需要提取出来用于汇总的，所以我们要使用 Python 实现读取 Excel 调查问卷的功能，最后再写入到一个新的 Excel 中。

好了，我们来看下这一功能的 Python 代码是如何编写的：

复制代码

```
import xlrd
import xlwt
from pathlib import Path, PurePath
# 导入excel和文件操作库

# 指定要合并excel的路径
src_path = '/Users/edz/Desktop/文章1/调查问卷'
# 指定合并完成的路径
dst_file = '/Users/edz/Desktop/文章1/result/结果.xlsx'

# 取得该目录下所有的xlsx格式文件
p = Path(src_path)
files = [x for x in p.iterdir() if PurePath(x).match('*.xlsx')]

# 准备一个列表存放读取结果
content = []

# 对每一个文件进行重复处理
for file in files:
    # 用文件名作为每个用户的标识
    username = file.stem
    data = xlrd.open_workbook(file)
    table = data.sheets()[0]
    # 取得每一项的结果
    answer1 = table.cell_value(rowx=4, colx=4)
    answer2 = table.cell_value(rowx=10, colx=4)
    temp = f'{username},{answer1},{answer2}'
    # 合并为一行先存储起来
    content.append(temp.split(','))
    print(temp)
    # 输出
    # 韩梅梅,D,B
    # 李雷,D,C

# 准备写入文件的表头
table_header = ['员工姓名', '第一题', '第二题']

workbook = xlwt.Workbook(encoding='utf-8')
xlsheet = workbook.add_sheet("统计结果")

# 写入表头
row = 0
col = 0
for cell_header in table_header:
    xlsheet.write(row, col, cell_header)
    col += 1

```

```
# 向下移动一行
row += 1
# 取出每一行内容
for line in content:
    col = 0
    # 取出每个单元格内容
    for cell in line:
        # 写入内容
        xlsheet.write(row, col, cell)
        # 向右移动一个单元格
        col += 1
    # 向下移动一行
    row += 1
# 保存最终结果
workbook.save(dst file)
```

在这段代码中，Excel 的读取和写入操作、for 循环操作都派上了用场，它的整个工作过程就像我画的时序图一样：**先打开用来汇总的 Excel 文件，依次对多个调查问卷进行读取，最后逐行写入到新建立的汇总文件中。**

合并后的效果如下图：

员工姓名	第一题	第二题
韩梅梅	D	B
李雷	D	C

有一点需要你注意的是，为了让你更好地理解 Python 循环的工作过程，我没有严格遵守编码规范，而是尽可能地让程序按照从上到下的顺序执行。如果你已经不是第一次使用 Python，可以根据你的使用习惯，将程序进一步封装成 Python 里的函数，更加优雅地去实现它。

总的来说，在使用 Python 对 Excel 的合并操作时，需要你掌握的最核心的操作就是读写文件和行列坐标控制。熟练掌握这两个功能，你才能一次得到想要合并成的样子，不用再对 Excel 进行二次的手工操作。

怎样实现 Excel 的拆分？

对于批量操作 Excel，还有一种情况是批量拆分。比如很多公司会用 Excel 记录和统计员工的薪水、记录货物信息、记录客户情况等数据。这些数据越来越多之后，文件会越来越大，打开文件和查找速度就会变得很慢，最后只好按照某些列进行 Excel 的拆分。

接下来，我就为你讲解一下如何进行 Excel 的批量拆分。让我们来看一个工资条的案例。

例如我在一个 Excel 中存放了工资信息，需要把第一行的表头和员工工资拆分成一个以员工名字命名的 Excel 文件。我来带你看下具体该怎么操作：

员工编号	姓名	出勤	工资	奖金	职务工资	全勤奖	实发工资
1001	韩梅梅	30	2000	1000	1000	100	4100
1002	李雷	30	2500	1000	500	100	4100

如果把拆分工作也画成时序图，就会发现，逐行读取可以使用循环功能批量操作，对每一行的内容处理，如果能使用 Python 进行自动化的话，一个 Excel 拆分的工作就全部能使用 Python 自动化实现了。所以，我打算设计一个 for 循环语句用于遍历所有的行，在 for 循环语句当中实现对每一行具体内容的处理。

我把文件拆分的关键代码写了出来，你可以参考一下：

复制代码

```
for line in range(1,employee_number):
    content = table.row_values(rowx=line, start_colx=0, end_colx=None)
```

```
    # 将表头和员工数量重新组成一个新的文件
    new_content = []
    # 增加表头到要写入的内容中
    new_content.append(salary_header)
    # 增加员工工资到要写入的内容中
    new_content.append(content)
    # 调用自定义函数write_to_file()写入新的文件
    write_to_file(filename = content[1], cnt = new_content)
```

在这段代码的第一行，我使用了一个 range 函数，它会生成从 1 到员工总数的数字范围。你可能会问，为什么没有直接写出 Excel 中总员工的数量，而是使用 employee_number 这样一个变量呢？

这是因为，如果直接写出员工数量，一旦遇到员工入职或员工离职等情况，你就需要根据 Excel 中的行数重新编写 Python 代码，而我现在使用的方式是每次打开 Excel 文件，会自动统计员工的数量（即行数），这种编写代码的方式能够让你的程序有更好的扩展性，一般这种方式用于处理文件内容经常变动的情况。

文件的批量拆分也是通过循环来实现逐行处理的功能的，但是你需要注意拆分以后的要保存的文件名称不要重复，不然很容易导致 Excel 中只有最后一次循环写入的内容。

小结

今天，我为你讲解了如何使用 Python 代替手工进行 Excel 的批量合并和拆分操作。一般来说，批量合并和拆分主要有几个步骤：

1. 手动或借助时序图找到需要重复操作的部分。
2. 将重复的部分用 Python 实现自动化操作。
3. 对 Python 实现自动化的脚本，再用循环功能实现批量操作。

Python 有着丰富的扩展库，**当你掌握了熟练操作 Excel 的方法之后，对于 WPS 等其他办公软件，也可以通过相应的 API 进行类似的合并和拆分操作。**

除了基本操作呢，我还给你介绍了分析问题的方法——时序图。通过时序图，你可以找到那些运行逻辑中重复的部分，利用循环完成自动化操作。

不管是利用 Python 代替手工操作节约时间，还是使用循环代替手工多次执行减少工作难度，这些都是提升工作效率的有效办法。希望你能在处理日常工作时多思考，有意识地把你的日常办公工作自动化。

思考题

欢迎你在课程后留言，告诉我使用 Python 解决了你工作中的哪些重复性问题。

如果你觉得这节课有用，能解决你的办公效率问题，欢迎你点击“请朋友读”，分享给你的朋友或同事。

提建议

© 版权归极客邦科技所有，未经许可不得传播售卖。 页面已增加防盗追踪，如有侵权极客邦将依法追究其法律责任。

上一篇 导读 | 入门Python的必备知识

下一篇 02 | 善用Python扩展库：如何批量合并多个文档?

精选留言 (20)

写留言

达文西 置顶

2021-02-03

上周刚碰到类似的需求,客户发过来的excle跟系统要求的模板跟数据格式都不对应,上万条数据,手动改肯定是搞不来了.就自己摸索着一天用python处理了一下做适配.不得不说python确实简单好学,很适合处理这些简单重复的业务场景.

展开

作者回复: 这是python最擅长的领域之一了。给这些非标的数据做格式化特别好用。

6

旧草
2021-02-04

例子保存的excel是xlsx，而官方文档写着：
xlwt is a library for writing data and formatting information to older Excel files (ie: .xls) by https://xlwt.readthedocs.io/en/latest/

作者回复: 感谢指正，xlsx应该为xls格式

1 5

Fan
2021-02-03

哈哈哈，这个主题讲得好，比爬虫，Django 那些内容实用多了。

作者回复: 都很有用，过年女朋友让你在某东某宝抢个化妆品，爬虫的功能就派上用场了。

3

李京斌
2021-02-07

能否共享讲课中提到的文件（EXCEL文件等）。

展开

作者回复: https://github.com/wilsonyin123/python_productivity
代码和课程中的文件都在链接中

2

Nick
2021-02-04

老师，我现在正好也碰到类似这样合并Excel文件的业务场景。大概有几十个拆分的Excel文件，要合并导入到数据库中，按照今天课程的思路，将所有的数据合并到一个Excel文件中来处理，但现在的问题是数据量比较大，超过了千万条记录。已经超出了Excel的最大范围，请问这种情况该怎么处理？我自己目前想到的是通过循环读所有的Excel文件，将数据一条条的插入到数据库中，请问老师还有什么高见？

展开

作者回复: 千万条数据存入到 Excel已经不合适了。存入到数据库是比较好的解决方案。
我们以MySQL为例，为了避免你的内存不足，建议你读取一个文件向数据库写一次。python可

以使用mysqlDB这个库写入数据库。
如果你需要跨文件做一些关联处理，必须一次性把这些文件都读入到内存的话，可以使用一个叫做pandas的库来进行相关处理。
如果数据的量级再大，建议你直接写入数据库，用SQL来处理数据之间的关系。

1 3

巩春雨
2021-02-08

老师，您讲的都是对于一些简单格式的excel数据有效。实际工作中财务的一些表格都是包含大量的格式，如合并单元格，涂色，隐藏，筛选及单元格内存在大量的公式。如果我把10个文件合并到一个文件中的不同sheet，如何能把相应的格式及公式快速的复制过去

展开

作者回复: 你好啊, 我想先带你解决从手动到自动的思路问题, 自动化的前提解决了,再细化特殊问题,.
比如你提到的,单元格公式问题,公式能够正常处理的前提是是否被正常读取了,你会发现openpyxl、xlrd这两个模块在读取excel的时候,有一个data_only 参数,控制着读取公式的行为, data_only为True读取的公式单元格就是None(需要用excel再打开文件一次才能显示值),如果是False就是公式本身.
还有一个终极方案,用python将公式内容以python字符串方式写入,如:
xls["B10"] = "=SUM(B1:B9)"
通过这两个参数控制

1 1

落曦
2021-02-07

老师您好，我试了一下您的代码，第一个有问题，运行出错：
Traceback (most recent call last):
File "D:/pycharmproject/Exceldemo.py", line 10, in <module>
value = table.cell_value(rowx=4, colx=4)
File "D:\pycharmproject\venv\lib\site-packages\xlrd\sheet.py", line 420, in cell_v...

展开

作者回复: 您好, 错误提示如下:
IndexError: list index out of range
是在
value = table.cell_value(rowx=4, colx=4)
出现了列表超过索引范围报的一个错误.

为什么报错的是一个列表呢?cell_value的底层实现是一个列表,所以报错提示是列表的索引超过了限制,

什么时候会报错index error ?比如列表只有三个元素,你引用第四个自然会报错,如果再往前面的代码追溯,就是文件打开失败,导致了列表没有数据,而你读取了_cell_values[4][4],所以超过了列表的长度限制,报了错

中间运行过程你可以了解一下, 你的解决办法是对的,从文件名下手.更正了文件路径错误的问题.后续我还会为你讲解异常处理,你可以在打开文件失败的时候就直接报文件打开失败错误,就不用通过内置错误间接分析文件名指定错误的问题了

1

Bryan
2021-02-05

老婆总是让我批量处理文件，上次学用 PowerShell 写了一个小程序，非常难写。用Python 简单多了。

作者回复: Python在网络和文件库的丰富程度上，是ps无法比拟的，但是在命令行方面ps也是非常强悍的，我在加餐当中也会使用到powershell，为你讲解它擅长的地方。

1

湘
2021-02-11

老师，我从github上下载您的代码是乱码，请问我这边需要做什么操作才能看到代码呢？谢谢。

作者回复: 如果是windows系统可以考虑使用sublime vscode等软件打开代码，因为我是使用mac系统建立的文件，默认是utf-8的文件编码，而windows是gbk的，所以会有乱码出现

琥珀•
2021-02-10

import xlrd
import xlwt

def get_table(src_file):...

展开

作者回复: 赞

乐途

2021-02-10

import xlwt
import xlrd

"""

对工资表按照个人进行拆分，然后批量保存为按照员工姓名命名新的独立表格...

展开

乐途

2021-02-10

拆分部分比较简略，我贴下我自己敲的，望老师指正：

import xlwt
import xlrd
...

展开

作者回复: 因为面向初学者，包括我的演示代码写的也比较简单，我认为需要从几方面优化代码，第一个是要操作的文件和输出路径要通过命令行参数处理，第二个是变量的名称要有一定的含义，最后是可以考虑继续抽象和封装代码里的其他功能，除了write-to-file外，其他逻辑看有没有能复用的，整体上完成的还是比较棒的

1

哒啷

2021-02-08

无法读取xls或xlsx出现错误可能是由于xlrd版本过高，输入pip3 install xlrd==1.2.0即可

作者回复: 赞, 我再提供一个库openpyxl, 它的社区呼声更高一些,了解完xlrd可以对比学习一下

McKee Chen
2021-02-07

老师可以把代码链接放在文章里吗~谢谢老师

展开

作者回复: https://github.com/wilsonyin123/python_productivity

我调整一下链接,让你在后续内容更容易找到代码和样例文件

杏林之虎
2021-02-06

待拆分Excel文件含有18位的数字信息，读取的时候正常，但写目标文件的时候因记录方式丢失后面几位。通过api.NumberFormat=“@”，可以解决，但处理文件运行速慢，上万条数据的话，就会死掉。请问有什么办法可以解决嘛?

展开

作者回复: 18字段,上万行内存消耗会占到5GB以上,内存的巨大开销让你的运行速度过慢,主要解决办法是需要减少内存开销

python支持
excel写入的库很多,对大内存支持比较好的一个是xlsxwriter,使用方法:
xlsxwriter.Workbook(file, {'constant_memory': True}),它的参考文档是https://xlsxwriter.readthedocs.io/
试试改为这个库能否加快文件存储速度

lynn
2021-02-05

可执行文件怎么获取啊

展开

作者回复: 再详细描述一下你的问题?

1

栾~龟虽寿!
2021-02-05

老师，有读一行的函数了，

展开

作者回复: 好，我去看一下

廖向冰
2021-02-04

不是视频课?

展开

作者回复: 文字专栏

LM
2021-02-04

拆分的讲解有点缩略了，比如我想把一个excel里的特定行、列内容拆分出来，再写入到一个新excel的特定行列，这个怎么做

展开

作者回复: 通过下面的方法可以实现
SheetObject.row_values(rowx[, start_colx=0, end_colx=None])

取sheet中第rowx+1行，从start_colx列到end_colx列的数据

返回值为列表，但是写入的时候，因为操作的基本单位是单元格。你需要通过两个for循环进行行列控制进行写入。

还有一种写作方法就是使用pandas库。把读出来的。多行多列的嵌套列表，写到新的Excel当中。

除非你有非常多的格式相同的，Excel需要这样处理，否则Python在这种场景下起不到优化效率的效果。反而让你在维护Python脚本的工作当中，浪费更多精力。

在野

2021-02-04

有完整的代码吗

展开

作者回复: https://github.com/wilsonyin123/python_productivity

1